DATA MINING

Field of the Invention

The present invention relates to a multi-agent system, which in particular, but not exclusively, is concerned with organising data records, e.g. searching for data patterns amongst a plurality of data records.

Background of the Invention

It is often desirable to understand patterns in data records, in particular where there is a larger number of computer-managed records. The problem of data clustering is well understood and more or less solved for the cases when all data that is to be analysed is known in advance, i.e. in a situation where the data records do not change during analysis – a so-called static environment.

There are however many important situations where data arrives for analysis in small batches at frequent, unpredictable intervals. Perhaps the most interesting example is an Internet portal with a large number of visitors who leave behind a small but significant amount of data whenever they visit the site. To extract a coherent and up-to-date pattern of behaviour of customers, it is desirable to ensure that the clustering process is dynamic, that is, capable of taking into account data as it arrives. Current clustering algorithms cannot cope with these conditions.

Although search engines are known which are capable of providing an analysis of a historic database, these search engines do not adequately cope with the demands of a randomly changing database, in which data records are received frequently and at random. In such databases it is necessary to freeze the set of data at some point and perform a historical analysis of this data.

Existing programs for data analysis rely on the statistical measurements etc of static sets of records.

In these systems the data in the database is frozen at any point in time so that the database can be analysed.

It is an aim of the present invention to locate data patterns in a database that changes dynamically.

In other computing areas, in particular simulations of a real environment by a virtual environment, the growth of so-called "agents" has increased. An agent, as described herein is a software object capable of communicating autonomously and intelligently with other objects. Any system can be modelled by identifying the key elements of that system and then developing specialised agents that have the same properties and attributes representing those key elements.

Agents are known in the context of managing a virtual environment from GB Application No. 0202527.8. In that application, an agent is an executable program which is capable of contributing to the accomplishment of a task and has the following features:

- Capable of accessing so-called "ontology" which contains domain-specific knowledge and general knowledge.
- Reasoning about its tasks.
- Composing meaningful messages.
- Sending them to other agents or humans.
- Interpreting received messages.
- Making decisions based on available information plus the domain knowledge.
- Acting upon decisions in a meaningful manner.

A virtual environment can be constructed from a plurality of agents which interact with one another to simulate a real life scenario. In particular, this document describes a virtual resource market in which agents negotiate with one another with a view to achieving the optimal, or at least a satisfactory, allocation of resources to demands. That is, agents are assigned to each demand and resource which forms part of the virtual resource market.

According to a first aspect of the present invention there is provided a data agent for organising data records, the data agent representing a data record and comprising: an agent descriptor implemented as an executable program and comprising a set of record parameters defining the type of data record it represents; and an agent body implemented as an executable program and comprising a negotiating interface for communicating with other agents representing data records; and a decision engine operable to determine when a record is a match for the type of data record represented by the agent based on a cluster valuation formula and to form a cluster of the represented data record and the matching data record.

According to a further aspect of the present invention there is provided a cluster agent for organising data records in a system, the cluster agent representing a cluster of data records and comprising: an agent descriptor implemented as an executable program and comprising a set of record parameters defining the type of data records in the cluster it represents, with a cluster value representing the strength of the cluster; an agent body implemented as an executable program and comprising a negotiating interface for communicating with other agents representing cluster records; and a decision engine operable to negotiate with that agent and any other agents representing data records to determine if those data records should join the represented cluster according to a data valuation formula.

According to yet a further aspect of the present invention there is provided a computer system for searching for data patterns in a dynamically changing data store, the data store holding a plurality of data records, the computer system comprising: an agent creation means arranged to create an agent implemented as an executable program and which has a decision engine operable to match the represented data record with other data records based on a data valuation formula; and a sensor for sensing the arrival of a new data record at the data store and arranged to cause the agent creation means to create a new data agent; wherein the new data agent is capable of negotiating with any existing agents in the system to form a cluster of data records representing said data pattern.

According to another aspect of the present invention there is provided a method of searching for data patterns in a dynamically changing data store, the data store holding a plurality of data records, the method comprising: creating a new data agent for each new data record that arrives at the data store, the new data agent being implemented as an executable program and having a decision engine operable to match the represented data record with other data records based on a data cluster valuation formula; wherein the new data agent negotiates with any existing agents in the system to form a cluster of data records representing said data patterns.

A computer program product having program code means for implementing the method is also described.

A further aspect of the invention provides a method of operating a computer system to organise data records, the method comprising: sensing the arrival of a new data record at a data store adapted to hold a plurality of data records; instantiating a data agent as an executable program, the data agent representing the new data record; implementing a clustering process by causing said data agent to negotiate with existing agents, said existing agents including data agents for existing data records and cluster agents, wherein cluster agents represent a plurality of data records.

A further aspect of the invention provides a computer system configured as a multi-agent system to organise data records in a data store, the computer system comprising: a first set of data agents implemented as executable programs, each data agent comprising a set of record parameters defining the type of data record it represents; a second set of cluster agents implemented as executable programs, each cluster agent comprising a set of record parameters defining the type of data records in the cluster it represents; wherein the data agents and cluster agents are operable to negotiate by exchanging messages, messages from a data agent containing an application for membership of a cluster, and messages from a cluster agent including rejection or acceptance of the application, and wherein when a new data record arrives in the data store, a new data agent is created to represent the new data record and is able to disturb established clusters in such a way as to improve a system value representing the quality of clustering.

Where the agent comprises an agent descriptor and an agent body, the agent body can include a sensor which can comprise at least one of means for reading accessible data fields and a mail box mechanism for receiving messages.

The agent body may also comprise an actuator, which can comprise at least one of means for accessing a database to update data fields therein, and means for dispatching a message.

For a better understanding of the present invention and to show how the same may be carried into effect, reference will now be made by way of example to the accompanying drawings in which:

Figure 1 shows a basic structure of the hardware elements according to an embodiment of the present invention;

Figure 2 shows an agent architecture interfacing with the ontology according to an embodiment of the present invention;

Figure 2A shows a preferred embodiment of an agent architecture;

Figure 3 shows an agent scanning the database according to an embodiment of the present invention;

Figure 4 shows a flow chart indicating the clustering process upon arrival of a new data record according to an embodiment of the present invention;

Figure 5 shows an example of a dynamic data clustering for four data records;

Figure 6a-6d shows the clustering process for the example of Figure 5 when density is the cluster valuation criteria; and

Figure 7a-7c shows the clustering process for the example of Figure 5 when shape is the cluster valuation criteria.

The present system also uses a multi-agent system that is capable of self-organisation, i.e. the capability of a system for reorganise itself to achieve some optimal value without human intervention, and which is achieved by modifying existing and/or establishing new relationships amongst its agents, for a data mining application for searching for patterns of data in a database that changes dynamically. In particular, in the following described embodiments a unique architecture and approach for data mining is represented, with the following attributes, deliverable separately or together.

(1) an agent represents every data record and every cluster, (2) the dynamic creation of the networking structure of clusters, (3) negotiations initiated by either (or both) sides (clusters and records), (4) data elements are allowed to reconsider decisions and leave a cluster, (5) a unique decision making mechanism, (6) use of energy levels to determine clustering/agent management.

The benefits of such a system are far reaching, since it is possible to provide a more efficient data-mining tool for a database which changes dynamically.

Figure 1 shows a basic structure of the hardware elements according to one embodiment of the present application. That is, a server 10 is connected to a database 2 over a communication interface 8. The server and database are shown as receiving new data records along line 12. The server comprises programs which can create and execute agents. A plurality of agents 14 are shown schematically. Agents 14 can interact to form a multi-agent system and in particular are able to communicate and negotiate with one another autonomously to search for data patterns as described in the following.

The database is monitored by a monitoring program which continuously checks for changes in the data records. When a change is sensed, an agent is created for that new data record by the agent creation program. In addition, all existing agents are informed of the arrival of a new agent.

Once created, each agent comprises program code which is executable by one or more processors 120 that reside on the server 10. The server 10 is also shown to comprise a memory region 140. This memory contains the program for creating agents, the agents themselves and the ontology of the multi-agent system as shown in Figure 2. It should also be appreciated that in an alternative embodiment the database 2 can also be stored in the memory 140 of the server. As described more fully in the following, agents are created to represent data records and clusters of data records.

Figure 2 shows how an agent 14 is able to interact with the ontology 108 so as to autonomously reorganise itself and its relationship with other agents in the multi-agent system. An agent 14 is shown as comprising a descriptor 100 and an agent body 102. In the present system, the descriptor indicates the type of data record represented by the agent, for example a set of record parameters. Alternatively, the descriptor merely identifies a particular type of agent which then has a pointer 123 that points to the set of record parameters of the data record associated with that particular agent and which is stored in the ontology. For the example in Figure 2 the agent 40 is assigned to data record 2 and is able to access the relevant record parameters 120 stored in the domain-specific knowledge 110 part of the ontology 108. For the data clustering application, the specific domain will be a data record domain.

The ontology 108 also shows that there may be other domain-specific knowledge parts of the ontology 110' which might be applicable to different virtual environments. Also the ontology shows the general knowledge 114 part of the ontology 108. In one embodiment of the present invention the data record knowledge part 110 of the ontology comprises further intelligence for example clustering criteria 122 and rule sets 152.

The agent body 102 is shown to comprise a decision engine 104 and a scanner 106 which work together. The operation of the scanner 106 is to scan other data records and will be described in more detail later. The decision engine shows that a first line 125 connects to the general knowledge part 114 of the ontology, which allows the ontology to supply the necessary intelligence relied on by each agent. Also line 127 connects the decision engine 104 to the clustering criteria 122 of the data record knowledge part 110 of the ontology. Thus, in operation the decision engine 104 is able to determine whether a data record which is scanned in the database 2 by scanner 106 is a match for the type of data record represented by the agent itself, and if the match is suitable, depending on the clustering criteria, a data cluster is formed by combining the data record represented by the agent 14 and the matching data record in the database.

Figure 2A shows in more detail the components of the agent body 102. The agent body has sensors 20, actuators 22 and, optionally, a fact memory. The fact memory can form part of the ontology 108. As already described, the agent body includes the decision engine 104 and is connected to the main ontology 108. Examples of sensor elements 20 include timers, vision sensors, mail box mechanisms etc. Some of these interact with other agents and some with the real world. For example, a visual sensor is the mechanism that is used by agents to read the open data fields of an agent descriptor 100. Typically, the visual sensor consists of a software procedure and data structure built into the agent body or alternatively can be transferred to an agent upon request from a base class held elsewhere in the system. The vision sensor mechanism can also have filters which impose vision limitations so that only certain open parts of an agent descriptor can be read. The vision sensor is one means by which agents communicate.

The actuators 22 may include WAP phones, email and means of accessing an agent's database etc. Thus, these allow established relations to be implemented in the real physical world.

Figure 3 illustrates the scanning mechanism of each agent 14. That is, the agent 14 comprises a scanner 106 having a scanning index 109 which proceeds through the database 2 scanning each data record 4 of the database in turn. As each record is scanned, its parameters are compared with the scanning agent's set of parameters to see if it is a record of the same type, i.e. a matching record. Upon encountering a matching data record 4, the decision engine 104 of the agent is invoked to decide whether to form a cluster 6 combining the data record represented by the agent 14 and the data record 4' in the database. It is preferable to generate for each scanned record an associated agent (scanning agent), to be able to implement negotiations (and thus more complex logic) between the scanning agent and record\cluster agents. The comparison could be implemented in different ways, e.g., it could be based on the decision making mechanism of the decision engine, or it could be done by complex algorithms which take into account rules and properties stored in ontology, or even implemented as agent negotiations. Scanning can be implemented in different ways, and may require, for example, preliminary preparations, like creating hash tables, which allow faster location of data records that are "near". Figure 3 shows that the agents will scan through the database in a top-down approach proceding in the direction indicated by arrow 107. However, it should be appreciated that in practice such a scanning mechanism would be achieved by running a standard scripting language which identifies the various elements of a database, for example in tabular format, and then comparing these data records with that represented by the data agent 14 doing the scanning.

Therefore the multi-agent system is able to represent a virtual data market, wherein data agents can be assigned to data records and cluster agents can be assigned to clusters of data. A cluster of data is formed by grouping together two or more data records according to the clustering criteria 122. Figure 3 shows that the database 2 comprises a plurality of data records 4 and at least one cluster 6. The server 10 maintains a virtual data market which is an up-to-date representation of the database 2. That is, data agents are assigned to data records and cluster agents are assigned to clusters. Because of the high dynamics of the process it is not convenient to have agents permanently assigned to clusters. Cluster agents are thus preferably created with the cluster, and destroyed when cluster disappears. Data agents could be implemented in two ways – they could either exist all the time, or be created\loaded from the database when needed for negotiations. There could be various improvements to this process. For example, each agent could be allowed to save its history, that is, “acquired experience and knowledge” back into the ontology 108. The data agents and cluster agents which form part of the virtual data market in the server 10 are then able to negotiate amongst themselves the optimal, or at least a satisfactory, allocation of data records to clusters according to clustering criteria specified in the ontology. One way of proceeding with clustering is to carry out clustering until an overall system value 150 is optimised, this value being determined from individual cluster strengths.

In a particular type of data search, clusters of data are useful since they can be grouped together according to some characteristic or criteria. For example, a shop owner would be able to search all his data records for customers purchasing bread and milk on a daily basis. Such a data cluster is useful, for example, for stock planning purposes. Data records are grouped into clusters according to a set of given clustering criteria 122, which for example is shown in Figure 2 as being retained in the data record specific knowledge part 110 of the ontology 108. Thus data clusters are a good way to search for patterns in the data records. In the context of data clustering, data agents are given the task of deciding whether to join a cluster, and cluster agents are given the task of inviting other agents to join the cluster.

Each data element or record may be assigned an energy level which is measured in terms of agreed energy units (eu), which determines the ability of its associated data agent to search for an optimal cluster. The energy level can also be used to limit the time required to accomplish an acceptable clustering solution. In Figure 2, each data record is shown as having a corresponding energy level parameter 124 wherein the energy level for that data record is stored. The agent associated with a particular data record is then able to retrieve information as to the data record's energy level by accessing the data record knowledge part of the ontology. Thus, data records are forced to limit their searches for clustering opportunities to those clusters that are affordable bearing in mind the energy level of the data element. With unlimited energy levels, data records will continue to search for the optimal clustering solution even if the incremental increases in the overall system value are negligible.

The overall system value 150, can for example be stored in the general knowledge part 114 of the ontology and is the value of the overall clustering process of the virtual data market at any point in time. The guiding principle for the allocation of data records to clusters within the virtual data market is to maximise this system value 150. Normally, the clustering criteria 122 would specify at which point the maximisation can be stopped, which is normally at the point when the effort expended by the process of self-organisation yields incremental changes which are negligible to the system value.

The energy levels 124 attributed to each data record could be distributed equally, or by some other domain-dependent set of rules. These rule sets 152 are for determining the energy level for each data record and are shown in the data record knowledge part of the ontology. Various different models can be used to determine the rule sets. For example the energy levels could be distributed to data equally or alternatively in for example an e-commerce application, the energy level available to a data record can be set as a commission for each item sold. In this case, the data record corresponding to the sale of a batch of goods would be richer than a record for a sale of only one item of the same type. Therefore in general, data records are more important to users of given higher energy levels. This works as follows. Data records spend their energy by paying for their interactions (which for example include joining, forming and/or leaving) clusters. Important data records will have a higher energy level and can therefore enter into a greater number of clustering negotiations with a view to achieving an optimal clustering membership. In contrast, data records of lesser importance (i.e. those having lower energy levels) will be limited to a smaller number of clustering searches.

Each time a data record wishes to join a respective cluster it will expend a certain portion of energy and therefore data records having low energy levels will only be able to approach a small number of the data records in the database. On the other hand, those having a high energy level are able to approach all of the data records since this more demanding yet optimal clustering process is affordable to a rich data record (i.e. having a high energy level).

Clusters that attract more important data records (having high energy levels) accumulate large energy levels and are therefore more visible to the users. The visibility is a term used to denote that a particular result is presented to users in a way which will attract their attention. The clusters which represent most useful\confirmed dependencies and new knowledge are made “visible” to users to enable them to make decisions quickly and correctly. So the visibility is decided by the developers of the system and the main criterion is which result is the most interesting to users. That is why clusters with more energy are shown first. In this manner, it is possible for a user that is searching for a particular type of important cluster to be able to identify this more easily. This could be for example by way of a display or any other suitable user interface.

The cluster criteria 122, stored in the data record knowledge 110, is used by the decision engine 104 of agents 14 to decide which data records to join together. The clustering process relies on the negotiation between data agents and cluster agents and therefore each of these agents needs a clear decision making criterion for selecting the best possible clustering action. According to a preferred embodiment of the present invention there are two distinct clustering criterion involved in the decision making process. Firstly, the data agent uses a cluster valuation formula to decide whether to apply to join a particular cluster, and secondly the cluster agents use a data valuation formula to decide whether to accept the application made by the data element to that cluster.

More particularly, the cluster valuation formula specifies how a cluster value is to be assessed by data agents. In general, the cluster value depends on a number of factors including: the number of data records belonging to a cluster, the energy levels of the data records, the shape (or boundaries) of the cluster, the attributes of the cluster, and their variety. The shape of a cluster plays an important role in searching for certain geometric patterns, e.g. in a search for certain objects hidden in a scene. In such cases certain shapes will have a higher value than other. In one embodiment a simple and effective way of determining a cluster value is to equate it to the so-called "density" of the cluster. If the data records belonging to a cluster are represented in N-dimensions, where N is the number of attributes in the data set, we can define the cluster density as the number of data records within the cluster volume. The density represents the reliability of a cluster. The more records that belong to a cluster, the more reliable the cluster is considered, meaning that the dependency which cluster represents is proven by more records. For example, if we analyse data by selling goods in a supermarket and find that a lot of people simultaneously buy bread and milk, then the more people buy both bread and milk, the more reliable is our deduced rule that people who buy bread will also buy milk. Each new record that confirms this rule increases "density" of the cluster.

The data valuation formula specifies how a data record value is to be assessed by a cluster agent. In general, cluster agents want to maximise the value of their clusters when deciding whether to accept data records from an application to joint that cluster. The energy level of a data record usually plays an important role in this decision. If cluster density is to be maximised, then data records that increase the density will be preferred.

Clustering criteria are particular cases of negotiation rules. Whilst clustering criteria help to decide whether a record will join a particular cluster, negotiation rules may include all sorts of rules, e.g. the proper sequence of negotiations between different records and clusters etc.

Figure 4 shows a flow chart for the clustering process of the multi-agent system which uses self-organisation to keep up to date with a dynamically changing database. That is, as shown in Figure 1 new data records arrive constantly to be updated into the database 2.

Figure 4 shows an initial step S30, wherein a new data record is received. This data record is to be stored in the database 2 and will modify it. At step S32 the server 10 creates a data agent which is assigned to the new data record. At step S34, the new data agents invokes its scanning mechanism and trawls through the database 2 to consider available clusters 6. At step S36 the new data agent sends an application to join the clusters that are deemed appropriate based on the cluster valuation formula (which will select those clusters deemed by the user to be most attractive). Thus the data agent will apply for membership to clusters which satisfy this criteria.

At step S38 the relevant cluster agents receive the membership applications from the data agent and evaluate the application using a data valuation formula which specifies features of the data record that will be desirable to that particular cluster. Those cluster agents that decide that the new data record will increase the value of their cluster, will each send a membership offer to the new data agent, which is illustrated at step S42. If on the other hand, none of the cluster agents feel that the data record will increase their value, step S40 shows that no offers are made by each of the clusters.

At step S44 the data agent assigned to the new data record decides which cluster offers are suitable if there are a plurality of suitable offers, the data agents accepts the most suitable offer and joins that cluster. The most suitable cluster for a data agent to join is the cluster which increases the overall value of clusters in which the data agent participates. This can be decided in quite a simple way, that is by calculating overall "energy level" of all relevant clusters.

If on the other hand at step S44, it is decided that there are no suitable clusters available, the data agent will at step S48 attempt to form a new cluster with other data records in the database, which may or may not belong to existing clusters. This is achieved by sending cluster formation proposals to each of the data agents associated with these other data records as shown by step S52.

Step S54 shows that the data agents, to which formation of new clusters is proposed, consider the offer and accept it only if it increases the overall value of the system. By accepting the offer, the agents 14 effectively reorganise the whole virtual data system, wherein the previously established relationships between the released data records and their clusters are destroyed and new relationships between different data records are established which increase the overall value of the system. In other words, if the overall value of the system is increased, at step S58 a new cluster is formed wherein the whole system needs to be reorganised, i.e. self-organisation of the multi-agent system shown at step S62.

Step S60 on the other hand shows that if the overall value of the system is not increased, then no cluster is formed. In this case the relationships between agents do not need to be reorganised at this point. However, at step S64 a concept known as "taxing" can be introduced, based on the allocated energy levels.

In the context of data clustering in multi-agent systems, taxing can be used as one way to drop a so-called "out-of-date" data record from the clustering process. This evolutionary capability enables the multi-agent system to maintain the effectiveness of a dynamic clustering process over long periods of time. In one embodiment, the taxing is implemented as one of the rule sets 152 in the data record knowledge part of the ontology. For example a taxing model can be set up whereby data records pay a tax during their membership to clusters. A "tax" is a reduction in their energy levels. This model enables evolutionary changes in the virtual system because data records are forced to quit when they exhaust their energy levels and therefore leave vacancies for new data records. That is, the rule set could be such that it charges a set fee (in terms of energy) for each clustering process. For example a specific cluster might comprise a plurality of different data records each having their own energy level. Older data records that are attached to the cluster would have low energy levels and may no longer be relevant to the type of data being searched for. In this case when the energy level of an old data record is exhausted it is released from the cluster which encourages more relevant data records to join the cluster.

The taxing model encourages data records as well as clusters to consider their long term prospects when they make clustering decisions. For example a cluster that does not attract data records for membership, may decide to reduce its membership tax in order to encourage new members and thus prolong its life.

At step S66, it is shown that if taxing is employed and the data agent is not joined to a cluster it will over time eventually disappear as its energy level is eventually burnt out. On the other hand, step S68 shows that if no taxing model is applied, then the agent can exist indefinitely in the virtual world and will wait around until new agents arrive which hopefully satisfy its clustering criteria.

The taxing model is also indicated at step S50 in relation to the situation when a new data agent decides to join a cluster. However in this case taxing is performed on the cluster agent as opposed to the individual data agent. In practice, as explained above at step S51 older data records will eventually be taxed to such a low energy level that they will quit the cluster and the system once again enters into the process of self-organisation at step S62, wherein new relationships between the data agents and cluster agents need to be negotiated.

Step S70 indicates new agents have been created representing newly created clusters and/or clusters whose properties (value, boundaries, number of records) have changed during self-reorganisation at step S62. These newly created agents start a new negotiation round with selected data records, which is demonstrated by the line 71 which is fed back to step S34 so that the clustering process described above is repeated. The clustering process continues until all data records are linked to clusters and if no further change of cluster membership will increase the overall value of the system, or until the time for clustering is exhausted. Under conditions of perpetual arrival of new data records to the system, at some point in the clustering process agents will begin dropping out of data records from further clustering consideration if taxing is used.

Examples of dynamic data clustering will now be described to demonstrate how different cluster criteria 122 can lead to different cluster considerations.

Figure 5 shows an example having four data records, which arrive at the system one by one for data clustering. The data records are shown to be superimposed on a two-dimensional grid having an x and a y axis. This grid could for example be related to any database wherein a data record is logged in a two-dimensional coordinate system. For example, the four records indicated as: data record 1 at position (2, 4), data record 2 at position (3, 3), data record 3 at position (6,3) and data record 4 at position (7, 3). Assume that the cluster evaluation formula in this case is based on the density so that the association rule is “first consider the nearest data record or cluster”. In this case the clustering steps will be as follows:

1. Data record 1 arrives at the system.

2. Data record 2 arrives at the system. Data record 2 joins with record 1 to form a new cluster, which is shown as cluster 5 in Figure 6a.

3. Data record 3 arrives at the system and applies to cluster 5 for membership. However, cluster 5 rejects the offer made by data record 3, since its membership would reduce the cluster density. Data record 3 then suggests to cluster 5 to form a new cluster. They agree and form a new cluster shown as cluster 6 in Figure 6b.

4. Data record 4 arrives at the system and suggests to data record 3 to leave cluster 6 (in Figure 6b) and join instead data record 4 in a new cluster. Data record 3 agrees because the new cluster would have a greater density than cluster 6. Therefore cluster 6 is destroyed and cluster 7 is created from data records 3 and 4 as shown in Figure 6c.

5. Cluster 7 then proposes to cluster 5 to form a new cluster. They agree and form cluster 8 as shown in Figure 6d.

6. Cluster 8 realises that there are no further clustering opportunities available because all records and clusters have achieved their preferred memberships and the clustering process terminates.

In the second example, consider the same initial situation as shown in Figure 5, except that the decision-making criteria is slightly different. That is, in this example the cluster valuation formula will be based on the shape of the cluster (for example as used in standard pattern recognition techniques) rather than its density. Therefore in this example the negotiation rule is “consider data records falling into the same line”. Since the cluster valuation formula favours straight lines, the more records that fall onto the same line, the greater is the value of the cluster associated with that line. The clustering process steps are as follows:

1. Data record 1 arrives at the system.

2. Data record 2 arrives at the system. Since data records 1 and 2 are on a straight line they form a new cluster, cluster 5 as shown in Figure 7a.

3. Data record 3 arrives at the system. Data record 3 suggests to data record 2 that they form a new cluster as both records are on a straight line. Data record 2 agrees to join data record 3 to form a new cluster, represented by cluster 7 in Figure 7b. However it should be noted from Figure 7b that data record 2 still forms part of the original cluster 5.

4. Data record 4 arrives at the system. Data record 4 applies for membership to cluster 6 and is accepted since the membership of data record 4 increases the value of cluster 6 because it increases the number of points on a straight line (i.e. the cluster valuation criteria). Cluster 6 changes its boundaries and now incorporates data records 2, 3 and 4 as shown in Figure 7c.

It should be noted that both of these two examples illustrate simple clustering processes, wherein different cluster valuation formula lead to quite different results. It should also be appreciated that in this example the data records do not have any energy levels associated with them.

It is now useful to look at the various rule sets 152 (i.e. models) which are used by agents representing data records and clusters to negotiate cluster memberships with one another. These different rule sets or models can be set up depending on the searching criteria of the user of the system. For example two different models known as the so-called “club model” and the “shareholder model” will allow a prospective user to mine data from a dynamic changing database according to preferred requirements.

Broadly speaking the club model is a model where data records each pay a membership fee to join a cluster and these fees are fixed. In contrast, for the shareholder model, the data records buy shares in clusters wherein the share price is dependent on the number of data records that belong to the cluster and their energy levels (and these may vary in time).

Therefore in the shareholder model, data records have the opportunity to increase their energy levels by entering or quitting a cluster at an opportune time. The data records can also lose energy due to a wrong clustering decision. The shareholder model increases the differentiation between clusters from the point of view of their usefulness to users.

Therefore the club model commands equality and creates a larger number of clusters in the earlier stages because the membership fee is low and fixed and it is therefore easier to create a new cluster. Once in a cluster, data records are reluctant to change their membership because the low energy level of new records is insufficient to initiate the reorganisation of clusters. In contrast, the shareholder model provides elitism and differentiates clearly between rich and poor data records. Thus a smaller number of clusters is generated in the earlier stages, which enables a greater mobility of data records even in the late stages, because new rich data records can force the restructuring of clusters, even by ousting less rich data records from rich clusters. Moreover a higher speed of clustering is achieved because the number of options available to each data record is reduced considerably as a result of the high membership fee which prevents poor data records from joining rich clusters.

Finally, as it has already been described, a taxing model can be included in the rule set. The taxing model gives an additional dimension to the clustering process. That is, the data records have to pay fees to stay in the virtual system so that the structure of clusters will change with time as poor data records are forced to leave. In practice, the taxing model is normally used alongside either the club model or shareholder model to induce clustering evolution.

The process of data clustering and the results thereof depend on the selected model of cluster membership. In particular, the following cluster features are dependent on the clustering criteria:

- The size of clusters: a large number of small clusters or a smaller number of large clusters.
- Equality (all data records are of equal importance) versus elitism (some data records are given preferences).
- Speed of clustering

Details of the different cluster membership models will now be described using a physical example, in which the following data records exist:

Buyer's name	Goods purchased	Purchase value
John Smith	Beer	500
Bill Jones	Whiskey	13
Paul Gordon	Beer	10
Phil Bank	Beer	700
Ralph Leech	Vodka	20

Table 1

If the club model is used and cluster membership fee is set to 3 eu, and all data records are given an equal amount of money, say 10 eu, which is represented as an initial energy level, which in this example is equally distributed among data records. The system will generate the following two clusters:

Cluster Name	Cluster Members	Membership Cost	Cluster Energy Level
A Beer	John Smith Paul Gordon Phil Bank	3	9
B Beverages	John Smith Paul Gordon Phil Bank Bill Jones Ralph Leech	3	15

Table 2

To demonstrate the shareholder model, the same data records are considered, but now the shareholder model is applied for clustering. Assume that each data record has the amount of eu equal to the purchase value That is, in this example initial energy levels are not equally distributed to all records; the level depends on the purchase value. The amount of eu for membership in a cluster is calculated in the following way.

Consider first data record John Smith. That data record has the property that it can join with another data record to can form a cluster A, called Beer, for 10% of overall money (i.e. 50 eu). When the Phil Bank data record arrives he can decide whether to pay the same sum of 50 eu or a larger sum, say, 10% of his own overall money 70 eu. In the latter case he will receive a larger number of cluster shares, which could be sold at a later stage when the cluster becomes richer and his data record decides to leave the cluster and join another.

Therefore, the average cost for entering the cluster becomes (50+70)/2 = 60 eu.

In such a model it is not profitable for richer clusters data records which represent buyers with a great purchase value, to be shareholders of clusters with a small entrance fee and therefore having a small overall cluster value. Therefore, the following table emerges.

Cluster Name	Cluster Members	Membership Cost	Cluster Energy Level
A Beer	John Smith Phil Bank	60	120
B Beverages	Paul Gordon Bill Jones Ralph Leech	2	6

Table 3

Dramatic differences are noticeable when the clusters resulting from the shareholder model shown in Table 3 are compared to the clusters resulting from the club model shown in Table 2. The shareholder model clearly separates rich data records from the poor.

Thus, different models can be used depending on the requirements of the user, i.e. what data is to be searched for. The club model is useful if the user wants to know the kind of beverages that each customer (data record) purchases, whereas the shareholder model is useful if the user want to know which customers purchased certain goods in big quantities. Therefore based on the search results, the user can launch a focused advertising campaign that successfully targets the desired market sector.

The following example is used to illustrate the situation when the clustering criteria used, applies both the club and taxing models together.

In the example the initial conditions are that each data record has an energy level of 10 eu. The standard cluster membership fee is 3 eu, but could be changed by the clusters. The tax is 2 eu per clustering step.

Data Record	Purchases	Energy Level
John Smith	Beer	10
Bill Jones	Whiskey	10
Paul Gordon	Beer	10
Phil Bank	Beer	10
Ralph Leech	Vodka	10

Step 1:

Cluster Members	Cluster Name	Cluster Energy Level
John Smith (7) Bill Jones (7)	Beverages	6

Here the data records John Smith and Bill Jones form a cluster "Beverage" wherein each data record has 7 eu left (i.e. after the 3 eu membership fee has been deducted) and the cluster has an energy level of 3+3=6 eu.

Step 2:

Cluster Members	Cluster Name	Cluster Energy Level
John Smith (5) Bill Jones (5) Paul Gordon (7)	Beverages	7

In this step of the clustering process, the data record Paul Gordon joins the cluster and the first taxation is applied to the existing data records so that each data record (John Smith and Bill Jones) lose 2 eu. When the new record Paul Gordon arrived, it was decided that it was more profitable to join the existing cluster "Beverages" than to create a new cluster "Beer" with John Smith.

Step 3:

Cluster Members	Cluster Name	Cluster Energy Level
John Smith (3) Paul Gordon (5) Phil Bank (7)	Beer	9

It this step of the clustering process, a dramatic change has occurred. After the existing membership is taxed and the new data record Phil Bank arrived, it became more profitable for members to leave the cluster "Beverages" and form a new cluster "Beer". This was due to the taxation, wherein data records could not afford to belong to two clusters.

Therefore it can be seen that this type of model, which uses both the club and taxing models, generates a clustering process that places more emphasis on revealing up-to-the-minute trends as opposed to providing a more historical perspective.

The virtual data market and the intelligence of the cluster and data agents are able to offer an improved solution which is capable of taking into account newly arrived data records, and is able to able to adequately search for data trends in a dynamically changing database. The capability of the virtual data market to autonomously reorganise the relationships that exist between agents, for example by modifying existing relationships and/or establishing new relationships, allows for self-organisation of searching criteria for a dynamically changing database.